

INDEX

Symbols

- `/?` command, Ollama, 18
- `/clear` command, Ollama, 107
- `$and` operator, Chroma, 153
- `$eq` operator, Chroma, 153
- `$gt` operator, Chroma, 153
- `$in` operator, Chroma, 153
- `$lt` operator, Chroma, 153
- `$nin` operator, Chroma, 153
- `$or` operator, Chroma, 153
- `@app.post` decorator, FastAPI, 34
- `@dataclass` annotation, 148–149
- `@mcp.tool` decorator, FastMCP, 261
- `@tool` decorator, Hugging Face smolagents, 256

A

- A/B testing, 75
- accuracy of LLM responses, 39, 56–57
- Agent.ai platform, 249
- Agent Development Kit (ADK), 250
- agentic AI
 - autonomous agents, 249–254
 - and agentic workflows, 246
 - building first agent, 251–252
 - development environment setup, 250–251
 - and prompt engineering, 252–253
 - extending agents with tools, 255–268.
 - See also* MCP
 - building tools, 256–257
 - reusable tools, 257
 - tools with complex schemas, 265–266
 - writing effective tools, 257
 - workflows, 246
- AGI (artificial general intelligence), 7
- AI grounding, 8
- AI logic, 6–7

- AI workflow, 245–246
- all-MiniLM-L6-v2 model, 135, 143
- alternative indexing and retrieval, 183–184
- Amazon Web Services (AWS) MCP servers, 260
- ANN (approximate nearest neighbor), 142
- annotations, 34
- Apache Cassandra, 131
- APIs
 - API calls to LLMs
 - component overview, 32–33
 - FastAPI server creation, 31–32
 - FastAPI server setup, 33–34
 - launching and testing server, 22, 35–36
 - Node.js installation, 19–20
 - Ollama installation and local LLM setup, 17–18
 - project initialization, 19
 - sending prompts to local LLM, 18–19, 34–35
 - server file creation, 20–22
 - system requirements verification, 16–17
- ChatGPT as interface to, 12–13
- chat-style, 92–95
- hosted compared to Ollama, 178
- REST, 78

- `apply_chat_template` method, Hugging Face Transformers, 227–228
- approximate nearest neighbor (ANN), 142
- artificial general intelligence (AGI), 7
- assistant role, 94, 230
- attention mechanism, 236
- `AutoModelForSequenceClassification` class, Hugging Face Transformers, 217, 219

CONTENTS IN DETAIL

ACKNOWLEDGMENTS	xix
------------------------	------------

AUTHORS' NOTE	xxi
----------------------	------------

PREFACE	xxiii
----------------	--------------

INTRODUCTION	xxvii
---------------------	--------------

Generative AI and Beyond	xxviii
Data Security and Privacy	xxix
Technology Choice	xxix
Downloading the Example Code	xxx
The AI Coding Assistant Safety Recall Notice	xxxi
What We'll Cover	xxxi

PART I GETTING STARTED WITH AI

1	
UNDERSTANDING LARGE LANGUAGE MODELS	3

Understanding Text Generation	5
AI vs. Traditional Logic	6
The Versatility of LLMs	7
Understanding the Limitations	7
Limited Training Knowledge	8
Hallucinations	9
Model Bias	9
Multi-Step Reasoning	9
Improving LLM Responses	10
Prompt Engineering	10
Context Engineering	11
Fine-Tuning	11
Custom Models	11
Building Autonomous Systems	11
Plugging LLMs into Your Code	12
Summary	13

2 BUILDING YOUR FIRST LLM-POWERED APPLICATION 15

Calling an LLM by Using an API 16

- Verify System Requirements 16
- Install Ollama and Run a Local LLM 17
- Send a Prompt to the Local LLM 18
- Install Node.js 19
- Initialize Your Project 19
- Install Node.js Packages 19
- Create the Server File 20
- Launch Your Server and Test 22

Streaming Your LLM Responses 22

- Stream an LLM Response with Ollama 23
- Stream a REST Response with Express 23
- Install the CORS Middleware Module 24
- Create the Entire Server File with Express 25
- Launch Your Server and Test 26
- Stream to the UI with Fetch 26
- Decode the Response Chunks and Render 27

Summary 28

Exercises 28

3 PYTHON ESSENTIALS FOR LLMS AND APIS 29

Installing Python and Libraries 30

Calling an LLM via an API 31

- Create the FastAPI Server 31
- Understand the Components 32
- Set Up the FastAPI Server 33
- Send a Prompt to the Local LLM 34
- Launch Your Server and Test 35

Summary 36

Exercises 37

**PART II
PROMPT ENGINEERING**

4 FUNDAMENTALS OF PROMPT ENGINEERING 43

Programming vs. Prompting 44

A Clear Use Case for an LLM 45

Evaluating Your Use Case	48
Prompting Basics	48
Prompt Elements	50
Context Windows	51
Tokenization	52
Avoiding Runaway Costs	55
Accuracy and Performance	56
Choosing a Model	57
Summary	60
Exercises	60

5 PROMPT ENGINEERING TECHNIQUES 61

Instruction Prompting	62
Data Extraction	62
Personally Identifiable Information Removal	63
Synthetic Data Generation	65
Persona Prompting	66
Zero-, One-, and Few-Shot Prompts	68
Chain-of-Thought Prompting	72
Prompt Chaining	73
Prompt Tips and Best Practices	74
Focus on the Positive, Avoid the Negative	74
Collaborate with the Model on Your Prompt	74
Treat Your Prompts Like Code	74
Experiment with Prompt Variations	74
Summary	75
Exercises	76

6 PROMPT ENGINEERING IN CODE 77

Choosing a Library	78
Connecting to a Hosted API	80
LLM Output Configuration	81
Output Length	81
Sampling Controls	83
Prompt Templates	85
Creating a Template with Jinja	85
Separating Prompts and Code with Jinja	88
Using Dynamic Prompt Templates with Jinja	89
Messages and Roles	91
Understanding Chat APIs	92
Including System Messages	95
Creating a Type for Messages and Roles	96

Conversation History	96
Maintaining Conversation History on the Client	97
Trimming the Conversation History	100
Summarizing the Conversation History	102
Choosing Conversation History Options	105
Structured Output	107
Defining a JSON Schema with Pydantic	107
Using Instructor	113
Extracting Data with Instructor	118
Prompting in Production	121
Request Retries	121
Guardrails and Injection	121
Token Counts and Usage Costs	123
Summary	123
Exercises	124

PART III VECTOR DATABASES AND RAG

7 VECTOR DATABASES IN PRACTICE 127

Helping Machines Understand Our World	128
Powering a Better Search Experience	129
Understanding Vector Embeddings	131
Choosing an Embedding Model	137
Dissecting Model Names	138
How Vector Database Retrieval Works	140
Using a Vector Database for Meaningful Search Results	142
Creating the Product Collection	142
Inserting Products into the Vector Database	143
Querying the Vector Database	146
Using a Vector Database to Make Recommendations	147
Defining the Data Classes	148
Retrieving the Recommendations	150
Identifying the Most Relevant Results	151
Filtering on Metadata	152
Summary	154
Exercises	154

8 DESIGNING A RETRIEVAL-AUGMENTED GENERATION SYSTEM 157

What Is Retrieval-Augmented Generation?	158
Connecting the Knowledge Base	160
Using Context Retrieval with a Vector Database	162

Understanding the RAG Pipeline	162
Setting Up the Environment	164
Chunking and Loading a Vector Database	165
Understanding Chunking Basics	165
Choosing a Chunking Strategy	167
Chunking the Documentation	169
Loading Chunks into the Vector Database	171
Completing the RAG Pipeline with an LLM	173
Creating the Prompt Templates	174
Prompting an LLM with Context	175
Creating the Service API	177
Improving Trust with Citations	178
Advanced RAG Topics	183
Alternative Indexing and Retrieval Strategies	183
Architectural Variants	185
Production-Level Refinements	186
Data Quality and Security	188
Summary	189
Exercises	190

PART IV ADAPTING MODELS TO REAL-WORLD TASKS

9 WHY AND WHEN TO CUSTOMIZE A MODEL 193

Building a New Model vs. Fine-Tuning	195
Fine-Tuning Strategies	197
Self-Supervised Fine-Tuning	198
Supervised Fine-Tuning	198
Reinforcement Learning Fine-Tuning	198
Fine-Tuning Methods	199
No-Code Fine-Tuning	199
Technical Fine-Tuning	201
The Fine-Tuning Process	201
Summary	202

10 PREPARING DATA FOR FINE-TUNING 203

Gathering the Raw Data	204
Structuring the Raw Data	205
Encoding the Labels	205
Splitting the Data	206

Creating a Labeled Dataset	208
Creating a Dataset	209
Creating the Train, Validate, and Test Datasets	210
Combining the Datasets	210
Verifying the Dataset	212
Fixing the Label	213
Summary	214
Exercises	214

11 FINE-TUNING MODELS IN PRACTICE 215

Choosing Models for Fine-Tuning	216
Establishing a Baseline	216
Fine-Tuning a Classification Model	217
Evaluating the Fine-Tuned Classification Model	221
Fine-Tuning an LLM	222
Why Not Prompt Engineering?	222
Introduction to PEFT and LoRA	224
Regular Fine-Tuning vs. PEFT	224
LoRA	225
Prompting an LLM	226
Understanding Instruction Templates	226
Customizing Generated Response Format	230
Controlling Where Generation Starts	231
Building the Dataset	231
Training the LLM Using LoRA	234
Reducing Parameters	234
Configuring LoRA	235
Training	238
Evaluating the Fine-Tuned Model	239
Summary	239
Exercises	240

PART V BUILDING AGENTIC SYSTEMS

12 FROM WORKFLOWS TO AUTONOMOUS AGENTS 243

What Are AI Agents?	244
Why Do We Need Agents?	244
Traditional Workflows	245
AI Workflow	245
Agentic Workflow	246
Summary	247

13		
BUILDING AN AUTONOMOUS AGENT		249
Setting Up Your Agent Development Environment	250	
Building Your First Agent	251	
Agents and Prompt Engineering	252	
Summary	253	
Exercises	253	
14		
EXTENDING AGENTS WITH TOOLS		255
Building a Tool	256	
Writing Effective Tools	257	
Building Reusable Tools	257	
Model Context Protocol	258	
MCP Concepts	258	
MCP Components	259	
MCP Transport Protocols	259	
Building an MCP Server	259	
Your First MCP Server	260	
Building a Special Calculator	260	
Defining Custom Tools	261	
Running the MCP Server	261	
Testing with Postman	262	
Integrating with Claude Desktop	263	
MCP Transports	265	
Building Tools with Complex Schemas	265	
Summary	267	
Exercises	267	
AFTERWORD		269
INDEX		271